

# Modelling and Designing of NLP Interface to Database for Afaan Oromoo

Shumet Walelign<sup>1</sup>, Debela Tesfaye<sup>2</sup>, and Teferi Kebebew<sup>3</sup>

<sup>1</sup>Lecturer, Department of Information Technology, IoT Hachalu Hundesa Campus, Oromia, Ambo University, Ethiopia

<sup>2,3</sup>Lecturer, School of Computing, JIT, Oromia, Jimma University, Ethiopia

Correspondence should be addressed to Shumet Walelign; shumetw549@gmail.com

Copyright © 2021 Made Shumet Walelign. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

**ABSTRACT-** Database management system has been widely used for storing and retrieving data. However, database is often hard to access the data since their interface is rigid in cooperating with user, due to that analysis of natural language query interface to relational databases has gained much interest in research community. This can be termed as structured free query interface as it allows the users to retrieve the data from the database without knowing the underlying schema. Structured free query interface should address majorly two problems. Querying the system with natural language interfaces is comfortable for the naive users but it is difficult for the machine to understand. The other problem is that the users can query the system with different expressions to retrieve the same information. The different words used in the query can have same meaning and the same word can have multiple meanings. Hence; it is the responsibilities of the NLI to understand the exact meaning of the word in the particular context to handle such problem the researcher develops an algorithm that is from manually crafted rule that extracted from user queries for proper query translation and SQL query generation. The proposed, pattern matching approach to natural language interface to database system has been passes through various phases. The idea behind the approach is to overcome the problem bottleneck lack of tools in the language, while pattern-matching method can fill the gap of those resources. This makes our approach suitable for AONIDB since Afaan Oromoo is under resourced language. The exact meaning of the word used in the query in particular context obtained using domain-mapping dictionary and database-mapping table constructed form student database that contains grade, department and course table for mapping database information with the help of an algorithm developed by researcher based on the rule that extracted from collected user query. The system has been developed in which user can input query in Afaan Oromoo language and can see result in the same language. The proposed system can handle list query, single and multiple conditional queries, aggregate function queries and join queries. The proposed system evaluated on 100 Afaan Oromoo queries using student database with precision, recall and F-measure. The result of testing is very encouraging and an average performance of 94.2% that implies that the systems validity and reliability is very high, an indicator of its strong and successfully features use and

operation. Yet; further experiments using different approaches that extend this work are needed for a better performance.

**KEYWORDS-** AONIDB, Afaan Oromoo Language Interface to Database, Natural Language Processing

## I. INTRODUCTION

Language is a medium of communication that enables human beings to exchange ideas and information [1]. In its written form it serves as a means of recording information and knowledge on a long term-basis and transmitting what it records from one generation to the next, while in its spoken form it serves as a means of coordinating our day-to-day life with others [1]. A set of written symbol and convention used by human beings for communication purpose called as natural language [2]. Languages, particularly natural languages studied by a discipline known as linguistics. Nowadays with the existence of computer technologies there is a huge effort going on towards processing of natural languages using computers. Academic discipline that study computer processing of natural language is known as natural language processing (NLP) or computational linguistics [3].

NLP is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for achieving human-like language processing capabilities. Natural Language Processing (NLP) is a field of research and application that determines the way computers used to understand and manage natural language text or speech to do useful tasks. NLP is a technique which makes the computer understands the languages naturally used by humans [4]. The term “natural” in the context of the language is used to distinguish human languages (such as Amharic, Afaan Oromoo, Wolayitta, Tigrigna, English, etc.) from computer languages (such as C, C++, C#, Java, and Prolog). The main advantage of NLP is that it relieves the burden of learning syntax [5]. It means there is no need to learn the database languages like SQL and no requirement of any special training before working with the NLP system. Natural language processing (NLP) is a field of study that deals with computing natural language with machine and provides a potential means of gaining access to

the information inherent in the large amount of text available through intranet and the Internet [6]. Natural language processing refers to human like language processing which reveals that it is a discipline within the field of artificial intelligence. However, the ultimate goal of research on natural language processing is to parse and understanding of language not fully achieved. For this reason, much research in NLP has focused on intermediate tasks that make sense of some of the structure inherent in language without requiring complete understanding. In the context of human computer interaction, there are many NLP applications [7] such as information retrieval systems, information extraction, speech recognition, language translator, question answering, natural language interface to database, dialog systems which are described in chapter two of this study. The most explanatory method for presenting what actually happens within NLP system is by means of the levels of language approach. There are different levels of NLP [3]: phonological (deals with the interpretation of speech sounds within and across words), morphological (deals with the componential nature of words), lexical (deals with the interpretation of the meaning of individual words), syntactic (focuses on analyzing the words in a sentence so as to uncover the grammatical structure of the sentence), semantic (determines the possible meanings of a sentence by focusing on the interactions among word-level meanings in the sentence), discourse (works with units of text longer than a sentence) and pragmatic (uses additional information about the social environment in which a given document exists [3]).

Database management systems used for storing and retrieving the data. Databases are very hard to access data since their interfaces are rigid in cooperating with users [8]. In relational databases, the data is stored using tables. The table contains set of rows and columns, each column represents attribute and each row represents the instance of the data for a set of attributes. The data manipulated using various operators with fixed set of keywords by following a set of syntax rules. By learning this structured query language one can extract the required data from the whole set of data, can also perform various operations such as update, manipulate and deletion of the data.

The relational database management systems are more popular based on the characteristics like its robustness and flexibility, high performance, scalability, data security and protection and flexible data maintenance [9]. Above all these advantages, it allows to index, perform aggregation, filtering and sorting done on the data using structured query language. Even though, RDBMS has those advantage it has some limitation to perform operations on the data, which is stored on databases, it is required to learn the structured query language. Since the naive user who knows only the natural language difficult to directly, access the required information from the databases. To come out from these limitations, it is required to design a tool that can understand the requirements of the naive user through natural language query, convert the natural language query into an equivalent structured language query. Then obtained structural query is used to access the required information from the databases. This kind of tool termed as natural language interface to databases or NLIDB system. Thus, the NLIDB take the input in natural language query, translate into structured query language, and returns the desired

information to the naive user. Database could store huge amount of data [9]. In order to query or retrieve information from a database by a novice user, a natural language that will provide correct and precise information without knowing the depth of structured query language is the need of the time. In this thesis, the researcher proposed the idea of using natural language interface in the context of database prompted the need for development of a system called Afaan Oromoo natural language interface to database. Afaan Oromoo is an Afro-Asiatic language, and the most widely spoken of the Cushitic sub phylum and widely used as both written and spoken language in Ethiopia and some neighboring countries [10] [11] [12] [13], including Kenya, Djibouti, and Somalia. Besides being a working language of regional government of Oromia. Afaan Oromoo is the instructional medium for primary and junior secondary schools throughout the region. Moreover, a number of literatures, newspapers, magazines, educational resources, official documents and religious writings are written and published in Afaan Oromoo. There are many design models/techniques proposed in the literatures in the field of NLIDB [5] such as pattern matching systems, syntax-based systems, semantic based grammar systems and intermediate representation of languages system.

#### ***A. Natural language interface to database***

NLIDB Systems is one of the earliest and most widely studied areas of natural language processing (NLP). It is the development of a natural language interface to database systems. Such front ends relieve the users of the need to know the structure of the database and offer a much more convenient and flexible interaction, because of this application of NLP is still widely popular today [5]. Natural language interfaces to databases systems allow a user to communicate with the database directly by entering the query in the form of a natural language question. The NLIDB system maps the natural language query to the appropriate SQL by processing the information in the query and correlation with the system and domain metadata. The purpose [14] of natural language interface to database system is to accept requests in Afaan Oromoo language and attempts to 'understand' them or we can say that natural language interfaces to databases are systems that translate a natural language sentence into a database query and then display result back to the user. NLIDB is a step towards the development of intelligent database systems to enhance the users in performing flexible querying in databases [15]. A complete NLIDB system will benefit us in many ways; anyone can gather information from the database by using such systems. Additionally, it may change our awareness about the information in a database. Traditionally, people are used to working with a form; their expectations depend heavily on the capabilities of the form. NLIDB makes the entire approach more flexible; therefore, it will enhance the use of a database. Although the earliest research has started since the late sixties, but in Ethiopian context there is some research conducted in NLIDB for Amharic but not for Afaan Oromoo. Natural language interface to database is a system that allows the user to access information stored in a database by typing requests expressed in some natural language.

## ***B. User Interface components (GUI)***

The client level presents a user interface to the user that manage interaction with the system which consists of several components, the components like a text field for the user to enter a natural language query and display output.

### *1. Linguistic component*

The module that controls the linguistic aspect, where the natural language Query is submitted a many analysis operation (morphological, syntactic and semantic analysis). At the end of this procedure, we obtain an intermediate logical query. It handles input as natural language, translate it into formal query and generate output in natural language from the result which comes after execution of query. In the first stage of linguistic processing, the natural language query is mapped and translated into the corresponding SQL query by appropriate mapping functions [16]. The linguistic component performs three analyses: morphological, syntactic and semantic explained as follows:

### *2. Syntactic Analysis*

The word syntax means grammatical arrangements of words in a sentence and their relationship with each other [17]. The objective of the syntactic analysis is to find the syntactic structure of the sentence. This splits the sentence into the simpler elements called tokens. Then the spelling checker check the token is correctly spell or not, or check the availability of tokens on the system dictionary. After the morphological analysis stage, the syntactic analysis is used to show how the words of query entered by the user are related to each another. This transaction will enable our system to know the syntactic structure of the request and explain the dependency relationships between different words based on metadata. This is done by applying a set of syntactic rules. These rules from user query, which describe the query structures of the request.

### *3. Semantic Analysis*

Semantic Analysis is related to create the representations for meaning of linguistics inputs. It deals with how to determine the meaning of the query from the meaning of its parts. So, it generates a logical query which is the input of database query generator. The overall objective of the semantic analysis is to assign a logical meaning to the syntactic analysis. This is done by applying a set of semantic rules, which are used to translate the syntactic analysis to a logical query. Each syntactic rule defined in sentences has a corresponding semantic rule. For this reason, the translation process is called rule-by-rule style [18]. This module is composed of a set of classes; each class contains multiple queries in natural languages that have the same logical interpretation.

### *4. Database component*

The database component consists of SQL generation and SQL execution. The SQL Generation component takes an Intermediate Query (IQ) as an input from Linguistic component and generates an equivalent SQL query as output. The SQL Execution component executes the generated SQL query and displays appropriate query output or message on GUI to user. It performs traditional database management functions. A lexicon composed of a number of tables that store natural language words and their corresponding mapping to formal objects that are used to

create a formal query. These tables can have entries of column name; attribute names, etc. questions entered in natural language translated into a statement with the help of lexicon. Then a formal query is formed by mapping these tokens into lexicon tables, which is executed and the result in natural language is given to user. Natural language database systems make use of syntactic knowledge and knowledge about the actual database in order to properly relate natural language input to the structure and contents of that database [16]. The database component consists of two parts: DBQ generation and DBQ execution.

## ***C. Technique used for developing natural language Interface to database***

### *1. Pattern-Matching Systems*

Pattern matching system is the earliest and the simplest techniques to implement natural language interface to database (NLIDB). These patterns and rules are fixed [19]. The rules states that if an input word or sentence is matched with the given pattern, the action has been taken. Those actions are also mention in the database [20]. The main advantage of pattern matching approach is that no elaborate parsing and modules of interpretation are required and the systems are very easy to implement.

### *2. Syntax-Based Systems*

In syntax-based system user questions are analyzed syntactically i.e.it is parsed and the resulting syntactic tree is mapped to an expression in some database query language [21] [5]. Syntax-based systems use a language system that explains the feasible syntactic structures of the user's query [22].

Syntax-based systems use a grammar that describes the possible syntactic structures of the user's questions. Syntax-based NLIDBs usually interface to application-specific database systems that provide database query languages carefully designed to facilitate the mapping from the parse tree to the database query. The main advantage of using syntax-based approaches is that they provided detailed information about the structure of a sentence. A parse tree contains a lot of information about the sentence structure; starting from a single word and its part of speech, how words can be grouped together to form a phrase, how phrases can be grouped together to form more complex phrases, until a complete sentence is built. Having this information, we can map the semantic meanings to certain production rules (or nodes in a parse tree). One of the examples of syntax-based system is LUNAR [23].

In this system grammar is nothing but the possible syntactic structure of the user's question. As Neeli Nihal ani [24] present the problem of syntax based, unfortunately not all nodes should be mapped, some nodes have to be left just as they are without adding any semantic meanings and it is not always clear which nodes should be mapped and which should not. Moreover, the same node in different parse trees is not necessarily going to be translated in all the trees. The second problem is a sentence can have multiple correct parse trees, and if all are translated, they may lead to different query results. The last problem is that it is difficult for a syntax-based approach to directly map a parse tree into some general database query language, such as SQL (Structured Query Language) [21].

### 3. Semantic Grammar Systems

In semantic grammar systems, the requests and responses are still done by parsing the input and mapping the parse tree to a database query [5]. The difference, in this case, is that the grammar's categories do not necessarily correspond to syntactic concepts [21].

The basic idea of a semantic grammar system is to simplify the parse tree as much as possible, by removing unnecessary nodes or combining some nodes together. Based on this idea, the semantic grammar system can better reflect the semantic representation without having complex parse tree structures. Instead of smaller structures, the semantic grammar approach also provides a special way for assigning a name to a certain node in the tree, thus resulting in less ambiguity compared to the syntax-based approach [22].

The main drawback of semantic grammar approach is that it requires some prior-knowledge of the elements in the domain; therefore, making it difficult to port to other domains [5]. In addition, a parse tree in a semantic grammar system has specific structures and unique node labels, which could hardly be useful for other applications. Much of the systems developed till now like LUNAR, LADDER, use this approach of semantic grammar.

### 4. Intermediate Representation

Due to the difficulties of directly translating a sentence into a general database query language using syntax-based approach, the intermediate representation systems as proposed by [22]. The idea is first to map a sentence into a logical query language, and then further translate this logical query language into a general database query language such as SQL.

## II. OVERVIEW OF THE PROPOSED SYSTEM

The first thing to be done with a user's query, is to tokenize the words in the user's queries into the words found in the dictionary and the requests tables of the database and if queries are wrong system auto correction is done to correct the query. This tokenization of words is done in such a way that erroneous repetitions are also eliminated, pre-processing phase briefly described in (section 4.4) of this chapter. In this study pattern matching approach with combination of the rule extracted from user query which are also used to develop an algorithm to understand the query feature of the sentences with the semantic feature learned from dictionary using matching rule. First, the manually crafted rule extracted from various queries provided by users of the query, and then an algorithm is developed based on those rules which are used for mapping and query translation purpose described in (section 4.14).

Since Afaan Oromo is under resourced language pattern matching approach is used in that pattern matching approach requires no elaborate parsing and modules of interpretation are required and Afaan Oromo lacks these readymade tools and pattern matching approach fill this gap and easy to implement. The approaches for query translation and execution are described below:

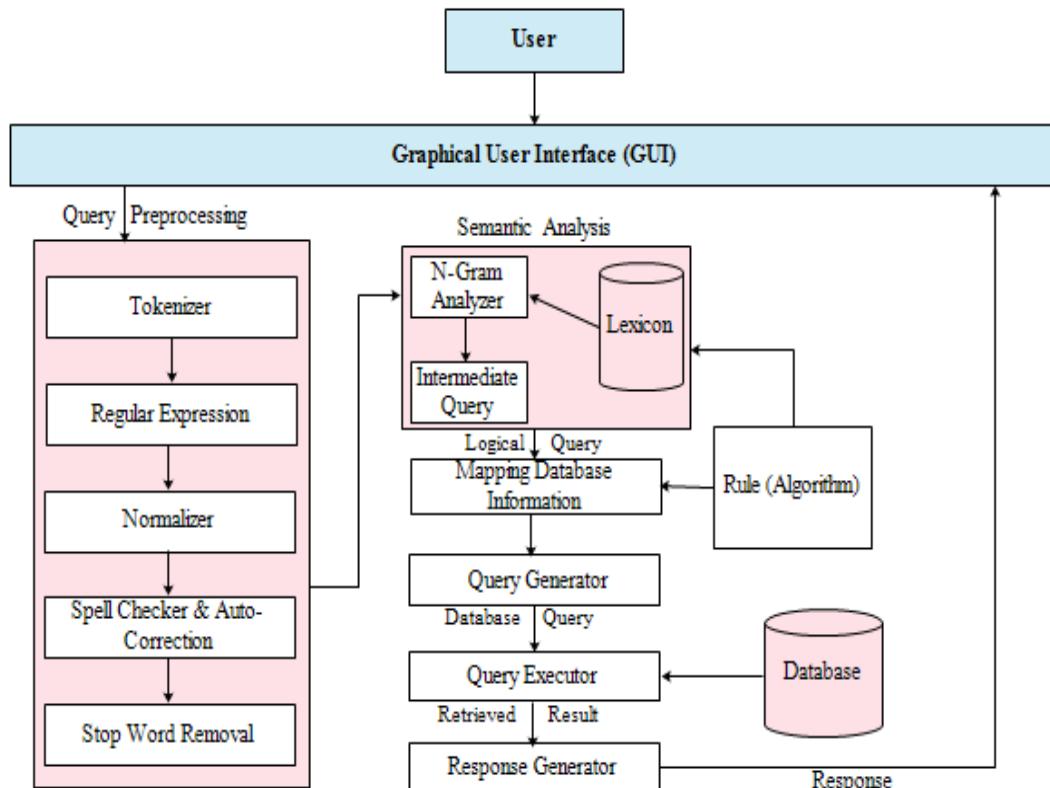


Fig. 1: General Architecture of the Proposed System

#### A. Constructing Rule for Developing an Algorithm

The linguistic knowledge of the language plays an important role to create the rule. The linguistic knowledge required for the natural language obtained in different ways. Identification of input query, whether the user query is table name, column name, column value, operand, and aggregate function or it may be noise word are the main task of the translation.

The rules to develop an algorithm extracted from the inherent structure of collected queries in forming meaning of words. The most common way of representing language is via rules [27] as cited in Worknight [28]. The rule

underlies many linguistic theories of the language, which turn into a set of rules. We have analyzed the structure of users query with respect SQL structure to develop algorithm. The rule is used to identify whether the user query is table name, column name, column value or position of table or column name which provide information and give more description about the query it used. These rules establish understanding of the structure of the database and enables retrieving of information from the underline database.

Table 1: Query Sample

No	User input query	System generated query	C/I/P
1	maqa barattootaa saala isaanii dubra ta'ee baasi.	select name from student where sex ='female'	C
2	maqa fi qabxi barattootaa saala isaanii dubra ta'ee baasi.	select name, gpa from student where sex ='female'	C
3	maqa qabxi fi umrii barattootaa saala isaanii dubra ta'ee baasi.	select name, gpa, age from student where sex ='female'	C
4	maqa qabxi umrii fi umrii barattootaa saala isaanii dubra ta'ee baasi.	select name, gpa, age, sex from student where sex ='female'	C
5	maqa fi lakk_barataa barattootaa GPA 3 qaban agarsiisi.	select name, sid from student where gpa =3	C
6	maqa fi umrii barattootaa GPA 3 qaban baasi.	select name, gpa, age from student where gpa =3	C
7	maqa fi umrii barattootaa GPA 3 olii naaf baasi.	select name, age from student where gpa >3	C

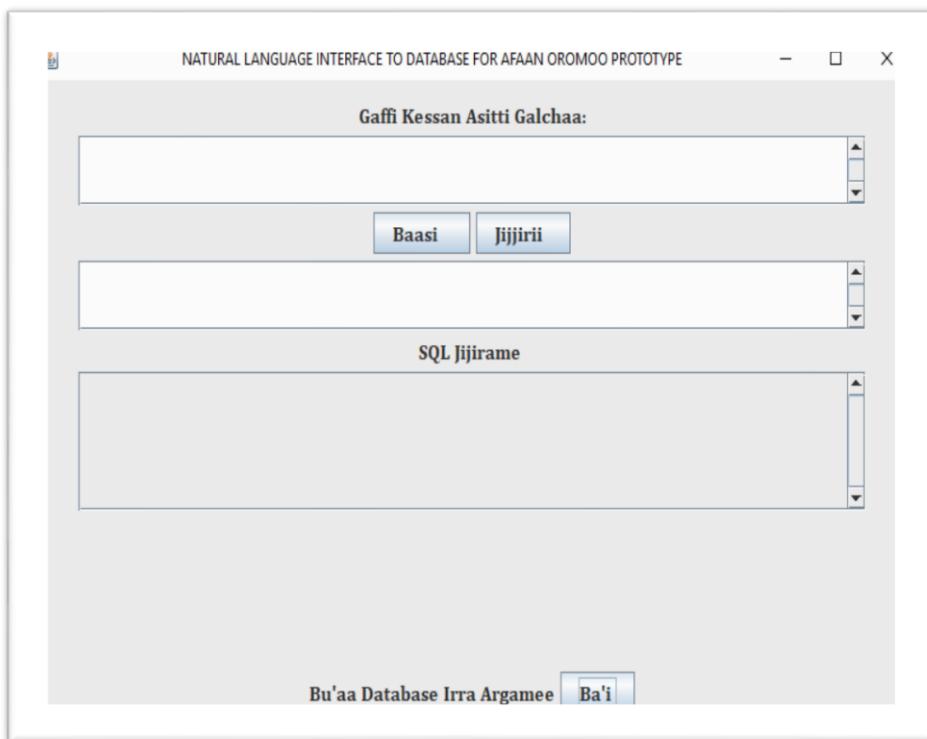


Fig. 2: Natural Language Interface to Database Design of Afaan Oromoo

### III. EXPERIMENTAL ANALYSIS

The flexibility of the natural language interface for relational databases is of great importance since it is almost unavoidable for users to make either typographical errors or input out-rightly wrong queries altogether [27]. Flexibility of an NLIDB also makes the computer appear intelligent. This is the main goal of, natural language processing, and attempts to make human computer interaction as easy as possible W. Woods does as cited in [27]. From the experimental results presented above, it is clear that the developed NLIDB is indeed flexible as intended. When we compare other previous types of natural language interface to database systems developed so far, our AONLIDB systems that we created shares some similarities, but also noticeable for unique feature incorporated. Obviously, in the previous sections on the evaluation of our system's performance, noticed that the AONLIDB system we have developed found to have high performance as evidenced by results of measurement of accuracy for all query type. One similarity that it has with other previous NLIDB systems, by other developers such as Preeti Verma et al, A. Kaur, B. Sujatha, et al, Smegnew A, and many others is that they designed a user interface to database system in a natural language, which is Afaan Oromoo in our case. In addition, intended to address database users' difficulties that have no knowledge of English and SQL; hence, it enables such users to use Afaan Oromoo language for their queries and access to the database. Alike other previously created NLIDB including the above ones, our NLIDB system also targeted at users' who have no any prior and expert knowledge and skill of how to operate or use a database. In our case, any novice user can make a use of, operate on database system whenever they need to access, and obtain the data they require. Another similarity it has is that, for instance, alike the NLIDB systems developed by Preeti Verma et al and A. Kaur, they proposed "Punjabi language interface to database", the design of AONLIDB system we have developed and implemented an algorithm to extract and map patterns of natural Afaan Oromoo query which is unique to our system only but they didn't. Whereas coming to the feature that makes our AONLIDB system different from other previous system; it has incorporated unique features and has constituted some important aspects that gave rise to its high performance as well as its simplicity and user-friendly nature. In the design of their database interface, different researchers have created algorithms to help them create, a platform where the database can understand queries in the natural language and convert it into SQL queries. In particular, to the AONLIDB system we have designed, we have also developed and established an algorithm specific to and that only works for the natural language of our database system. However, in different from such and other previous NLIDB system, we tried to overcome such limitations by developing unique algorithm for Afaan Oromoo, that did not exist previously and new in its type. This avoided any difficulty that our database could have faced in understanding Afaan Oromoo sentences, as result of character variations, punctuation and other lexical and semantic variations in the structure of the language. This gave rise to the simplicity and flexibility of our particular database system, to communicate easily with the database user without any language barriers. Hindi language

interface to database by Himani Jain translation of query in which that did not contains table name our system share this similarity but not their system shares automatic detection of names of column for column value. Generally, the proposed algorithm compared to other previous types of NLIDB systems developed so far, especially on local language Amharic by Smegnew A. Even thou we share some similarity our algorithm in depth try to overcome the limitation and the challenge that forwarded by researcher which automatic detection of column name given column value. In this study have an answer to this problem not only this but also their system recalls performance measure using two parameters indicates 91% but the proposed algorithm performance measure is 94.2%, using three different classes beyond this our system much flexible and user friendly than Simeone, this is the main goal of this study that also, the main challenging task of the study and which offers higher flexibility to user and it removes the burden of syntax from the user.

### IV. RESULT AND DISCUSSION

This is to say that we have evaluated our system from different dimension. Language word recognition: - the user can type the query without too much care of ambiguous and multiple forms of word such as "saala, korniyya, dhala, dubra, dubartootaa, shamarran" since, the system can understand such word in various alternatives. Relaxation of grammar: - The system can also understand queries of the user constructed in the form of various expressions or structured in various grammatical forms. For instance, the system can recognize the following two sentence forms of single query as "saala baasi or saala baratootaa baasi (show me sex of student/ give me student sex)". Language Understanding: - during the request, the user is not expected to include the column name on the query. For instance, in a certain query "dubra" express, the sex is female but our system would understand and should reframe the query as "saala isaanii dubra ta'ee baasi". Simplification: - The user can type the query without a need for complexity of expression and without the strong knowledge of SQL. We have collected to overcome the complexity sample queries from nonprofessional peoples those who have not knowledge structural database language. The accuracy of the system measured in term of precision and recall percentage with two classes that identifies query response as correct queries (C) and incorrect queries (I) partial queries (P).

Table 2: Query Response and Performance

S. N	Performance Factors	Selection Query (30)	Conditional Query (50)	Aggregate Query (20)	General Assessment (100)
1	Recall	0.9	0.9	0.75	0.85
2	Precision	0.964	0.937	0.937	0.942
3	F-Measure	0.93	0.918	0.833	0.875

### V. CONCLUSIONS

In this study, the researcher has proposed natural language interface to database for Afaan Oromoo language. The designed and implemented system is tolerant and cooperative to access data from the database, using student

database, the system that accepts query in Afaan Oromoo language and accepted query translated into SQL query, by mapping Afaan Oromoo word, with their corresponding English words from dictionary. Then SQL query executed on database to provide output to the user. This lets the novice user to formulate their queries in their native language without the knowledge of structural query language. As try to discuss in the previous chapters, in this study, pattern-matching approach for the development of AONLIDB recommended which was suitable for under resourced languages like Afaan Oromoo. In this study, the proposed system designed to handle the challenge in natural language query processing. The aim is to achieve correct SQL query translations for natural language query (NLQ). The interface developed uses pattern matching technique, to translate natural language query to SQL. It also uses set of production rule, which extracted out of user query to develop an algorithm. A series of steps like tokenization, lower case conversion, stop word removal, spell checker and auto correction, analysis of special token, checking word similarity, database element extraction, and SQL element extraction to convert natural language query to SQL Query. Any Interface with the use of natural language processing can be powerful enhancements, human language being so natural and considering Afaan Oromoo as in our paper, which is most commonly used language across the Oromia region. The goal of NLIDB is striving to answer users query correctly as intended by the user. The contribution of natural language processing in achieving the goal of NLIDB system clearly pointed out in our study. Furthermore, it has pointed out how NLP plays a significant role in enhancing the human computer's interaction by processing texts. To that end, natural language interface to database is one component of NLP contributing a lot to the effort of solving problem of accessing data from database that answers users question by translating natural language input into formal query. However, if the user enters unrecognized word and if those words are difficult to analyze their meaning due to overlapping domain then system may not be able to handle such queries or it may produce partial output. However, this work demonstrates how different NLIDB approach and methodologies in a specific domain using domain dictionary and other external resources. It also shows a method to assist user in database interaction by suggesting/providing auto correction words used queries from the system dictionary. In this study, the only focuses are on pattern-based interaction along with select, from, where and join clauses of SQL query and also handling complex query that results from the natural language query, group by, aggregate function, order by, clauses of SQL query can be done for handling these types of query. The validation of the prototype has shown encouraging performance. The results show that the system had improved F-Measure. However, there was minimal change in precision and recall, which can be resolved in the future work. The accuracy of the system measured in term of precision, recall F-measure percentage with three classes that identifies query response as correct queries, incorrect queries and partially correct queries. The results found are encouraging and the overall efficiency of system observed to be more than 94.2%.

## CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

## REFERENCES

- [1] Allen. pattern-based Language Understanding," 2nd Ed, California Redwood, Benjamin /Cummings Publishing Company, Inc. 1996.
- [2] Amanuel Alemayehu, "the Syntax of Interrogatives in Oromoo," Ma Thesis, Addis Ababa University, Addis Ababa, 2006.
- [3] Liddy E.D," Natural Language Processing," Encyclopedia of Library and Information Science, 2nd Ed. Ny. Marcel Decker, Inc 2001.
- [4] A. Kaur, "Punjabi Language Interface to Database," Thapar University, 2010.
- [5] Ashish Kumar and Dr. Kunwar Singh Vaisla," Natural Language Interface to Databases: Development Techniques," Elixir Comp. Sci. & Engg. 58 (2013) 14724-14727, on 17 May 2014.
- [6] Getachew Mamo, "Part-of-Speech Tagging for Afaan Oromoo Language," Addis Ababa, Ethiopia January, 2009.
- [7] Flickinger, D.(2015) Natural Language Engineering-Efficient Processing With Hpsg: Methods, Systems, Evaluation. At: <Http://Www.Coli.Uni-Sb.De/Nlesi/> Accessed at 3/15/2015.
- [8] Abhijeet R. Sontakke et al.," A Rule Based Graphical User Interface to Relational Database Using NLP," International Journal of Scientific Engineering and Research (Ijser), Issn (Online): 2347-3878, Impact Factor, 2014.
- [9] B.Sujatha, et al, "A Novel Architecture for the Natural Language Interface to Databases," International Journal of Computer Architecture and Mobility (ISSN 2319-9229) Volume 1-Issue 6, April 2013.
- [10] Oromoo Language. [Http://En.Wikipedia.Org/Wiki/Oromoo\\_Language/](Http://En.Wikipedia.Org/Wiki/Oromoo_Language;); Last Accessed on August 25, 20016.
- [11] Kekeba Tune, Vasudeva Varma and Prasad Pingali, "Evaluation of Oromoo-English Cross-Language Information Retrieval," Ijcai 2007 Workshop on Clia, Hyderabad, India, 2007.
- [12] Oromoo Language: Encyclopedia [Http://En.Allexperts.Com/E/O/Or/Oromoo\\_Language.Htm](Http://En.Allexperts.Com/E/O/Or/Oromoo_Language.Htm), Last Accessed on October 10, 2016.
- [13] C.Griefenew Mewis, "A Grammatical Sketch of Written Oromoo," Druckerei Franz Hansen, Bergisch Gladbach, Germany, 2001.
- [14] Jadhav Sneha, Raut Shubhangi and A.S. Zore, "Natural Language to Database Interface", International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 2, India, February 2014.
- [15] R. Kasper, "A Flexible Interface for Linking Applications to Penman's Sentence Generator," Darpa Speech and Natural Language Work- Shop, 1989.
- [16] Hanane Bais et al. "A Model of A Generic Natural Language Interface for Querying Database," I.J. Intelligent Systems and Applications, 2016.

- [17] A. H. Madessa, "Probabilistic Information Retrieval System for Amharic Language," Addis Ababa University, 2012.
- [18] A. R. Sontakke and P. A. Pimpalkar, "A Review Paper on Hindi Language Graphical User Interface to Relational Database Using Nlp," Int. J. Adv. Res. Comput. Eng. Technol., Vol. 3, No. 10, Pp. 3393–3397, 2014.
- [19] A. Kumar and K. S. Vaisla, "Natural Language Interface to Databases: Development Techniques," Elixir Comp. Sci. Engg, No. November, 2015.
- [20] I. Androutopoulos, G. D. Ritchie, P. Thanisch, and M. Road, "Natural Language Interfaces to Databases An Introduction," No. 709, Pp. 1–50, 1995.
- [21] Neelu. Nihalani, S. Silakari, and M. Motwani, "Natural Language Interface for Database: A Brief Review," Ijcsi Int. J. Comput. Sci. Issues, Vol. 8, No. 2, Pp. 600–608, 2011.
- [22] Preeti Verma et al., "Punjabi Language Interface to Database".
- [23] Smegnew A "Amharic Language Query Processing in Database Using Natural Language Interface," M.Sc. Thesis, School of Computing, Jimma University, Jimma, Ethiopia, August, 2008.
- [24] Fei Shao, Yanjiao Cao(2005) A New Real-Time Clustering Algorithm, Department of Computer Science and Technology, Chongqing University of Technology Chongqing 400050, China. Linguistics: Linguistic Studies in Honor of Jan Svartvik, London, Longman.
- [25] B. Sujatha et al." A Generic Model for Natural Language Interface to Database," IEEE 6th International Advanced Computing Conference, 978-1-4673-8286-1/16, 2016.
- [26] Joseph Weisenbaum, "Eliza Computer Program for the Study of Natural Language Communication Between Man and Machine," Communication of the Acm, Vol. 9. 1966.
- [27] Jurafsky, D. Martin, "Speech and Language Processing Pearson Education," 2nd Edition, 2009.
- [28] Workineh Tesema Gudisa, "Towards the Words Sense Disambiguation of Afaan Oromoo using Hybrid Approach," Jimma University Institute of Technology (Jit), October 3, 2015.

## ABOUT THE AUTHORS



**Mr. Shumet Waleign** is working as a Lecturer in the Department of Information Technology, IoT, AMBO University, Ethiopia affiliated to the Ministry of Science & Higher Education. He has experience of Teaching, Research and Administrative field. He has published a number of international journal papers related to the Information Technology. He has involved the Academic Professional Membership bodies and committee members.



**Mr. Debela Tesfaye** is working as a Lecturer in the Department of Information Technology, School of computing, JIT, Oromia, Jimma University, Ethiopia affiliated to the Ministry of Science & Higher Education. He has experience of Teaching, Research and Administrative field. He has published a number of international journal papers related to the Information Technology. He has involved the Academic Professional Membership bodies and committee members.



**Mr. Teferi Kebebew Tolassa** received his BSc degree from Addis Ababa University in Information Systems. After working as graduate assistant in Jimma University for 3 years. Once He again joined Addis Ababa University and did his Masters in Information Science. Currently, He is working in Jimma University, Faculty of computing and Informatics as a lecturer. Most of my research work is around natural language processing for Ethiopian languages. He worked on a number of university funded researches with spell checker for Afaan Oromo, spell checker for Amharic and Afaan Oromo information retrieval system.